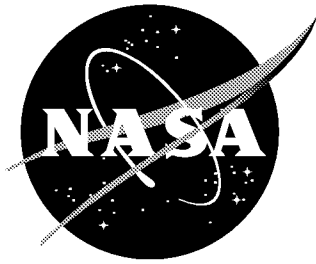


NASA/TM-1999-209711



Optimized System Identification

Jer-Nan Juang
Langley Research Center, Hampton, Virginia

Richard W. Longman
Institute for Computer Applications in Science and Engineering (ICASE)
Langley Research Center, Hampton, Virginia

October 1999

The NASA STI Program Office ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.

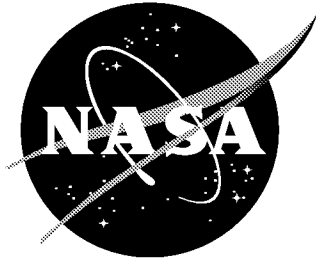
- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results ... even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at <http://www.sti.nasa.gov>
- E-mail your question via the Internet to help@sti.nasa.gov
- Fax your question to the NASA STI Help Desk at (301) 621-0134
- Phone the NASA STI Help Desk at (301) 621-0390
- Write to:
NASA STI Help Desk
NASA Center for AeroSpace Information
7121 Standard Drive
Hanover, MD 21076-1320

NASA/TM-1999-209711



Optimized System Identification

Jer-Nan Juang
Langley Research Center, Hampton, Virginia

Richard W. Longman
Institute for Computer Applications in Science and Engineering (ICASE)
Langley Research Center, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

October 1999

Available from:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 605-6000

OPTIMIZED SYSTEM IDENTIFICATION

Jer-Nan Juang¹

NASA Langley Research Center
Hampton, Virginia 23681

Richard W. Longman²

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, Virginia 23681

ABSTRACT

In system identification, one usually cares most about finding a model whose outputs are as close as possible to the true system outputs when the same input is applied to both. However, most system identification algorithms do not minimize this output error. Often they minimize model equation error instead, as in typical least-squares fits using a finite-difference model, and it is seen here that this distinction is significant. Here, we develop a set of system identification algorithms that minimize output error for multi-input/multi-output and multi-input/single-output systems. This is done with sequential quadratic programming iterations on the nonlinear least-squares problems, with an eigendecomposition to handle indefinite second partials. This optimization minimizes a nonlinear function of many variables, and hence can converge to local minima. To handle this problem, we start the iterations from the OKID (Observer/Kalman Identification) algorithm result. Not only has OKID proved very effective in practice, it minimizes an output error of an observer which has the property that as the data set gets large, it converges to minimizing the criterion of interest here. Hence, it is a particularly good starting point for the nonlinear iterations here. Examples show that the methods developed here eliminate the bias that is often observed using any system identification methods of either over-estimating or under-estimating the damping of vibration modes in lightly damped structures.

¹ Principal Scientist, Structural Dynamics Branch, M. S. 230.

² Also, Professor of Mechanical Engineering, Columbia University, New York, NY 10027

INTRODUCTION

There are many approaches to performing linear system identification from input-output data. One large class of such methods uses single-input, single-output higher order difference equation models as in [1], and makes a fit that minimizes the equation error. Another large class of identification methods create modern state variable realizations from data, as in the Observer/Kalman filter Identification (OKID) algorithm, see references [2,3] and its predecessors [4-6]. This approach makes a fit that minimizes the equation error of an observer, one which corresponds to a Kalman filter under appropriate limiting circumstances. In both cases, the minimization is not done with respect to what is really of interest in system identification for structural analyses. What one cares about is that the model developed has the property that when inputs are applied to the model, it produces outputs as close as possible to the corresponding outputs of the true system. Hence, it is the purpose of this paper to create system identification algorithms that are optimized with respect to this objective, i.e. that minimize the sum of the squares of the output errors of the model in fitting the data.

Optimization of this objective function is a minimization of a nonlinear function of many parameters. Here we use a nonlinear least-squares approach, involving sequential quadratic programming. Nondefiniteness in the second partials for the model sensitivity is handled by eigendecomposition followed by setting negative eigenvalues of the second partials term to zero. Because this is a minimization in a multidimensional space, it is possible to have the algorithm go to a local minimum, rather than a global minimum. Thus, it is important to start with initial conditions for the recursive process that are as close as possible to the true solution. Here, we choose to apply OKID to produce the initial starting conditions. OKID has proved very effective in difficult spacecraft identification problems [3]. We could also choose to use results from related algorithms. References [7,8] modify OKID to perform residual whitening, which can be helpful when the data set is relatively short. References [9,10] give modified versions of OKID that identify the Hankel matrix directly rather than the Markov parameters that occur in it.

OKID and these modified versions are good algorithms, which are often able to produce effective models in situation when other algorithms fail. The approaches developed here will enhance the results -- these approaches start with the OKID type result, and then minimize the model output

error. Whether a local or a global minimum is reached, the models produced will give outputs that are closer to the true system model outputs in the sense of the optimization criterion.

Here we develop algorithms to minimize the optimization criterion for five different types of system models:

1. Multiple-input single-output (MISO) ARX model or finite-difference model
2. A state variable model using an observable canonical form realization for MISO
3. A multiple-input multiple-output (MIMO) version of (1) above
4. A MIMO version of (2) above
5. A MIMO state variable formulation in block diagonal form for vibration modes

The last realization is particularly useful for structural dynamic systems or modal testing where one knows that the system is governed by vibration modes, and one is particularly interested in knowing the mode frequencies and the mode damping factors. These are being identified directly in this approach.

In the following sections we first develop the mathematical formulation of the problem, and then give the general form of the nonlinear least-squares algorithm in terms of the first and second partials of the model outputs with respect to the model parameters. Then we address each of the five types of models individually, developing analytical expressions for these first and second partials. There are, of course, nonlinear least-squares algorithms that do not require knowledge of the second partials, and there are algorithms that do not require knowledge of the first partials. For typical system identification problems, the run times are unreasonably long for such algorithms that do not require explicit expressions for these partial derivatives. Hence, one of the contributions of this work is to set up effective computation procedures for the first and second partials for each of the model types above, and this results in orders of magnitude improvement in run time, producing reasonable and practical algorithms. The approach has already been used in speech coding, with good results as described in [11].

Numerical studies are reported in the final section of this paper, which demonstrate that the procedures developed here give substantial improvement in system identification results. One

example of this is in the identification of the damping factors in lightly damped structural dynamic system, such as flexible spacecraft. OKID and Eigensystem Realization Algorithm [12-13] results very often overestimate the amount of damping in lightly damped modes. By applying the methods here to the OKID result, the bias in the damping is seen to disappear.

OPTIMIZATION CRITERION AND NOTATION

Data: Suppose that we are given experimental input and output data for N time steps, i.e. the input history $u(0), u(1), \dots, u(N)$ and the output history $y^*(0), y^*(1), \dots, y^*(N)$. These outputs can be considered to differ from the true system output due to noise in the data. Algorithms will be developed in the following sections for various types of models. Depending on the assumed model order, some of the beginning data points will correspond to initial conditions for the model, and then the first time step of the output of the model is denoted as time step $k = k_o$. Let \underline{y} be a column vector of the outputs starting with $k = k_o$ and going to $k = N$, giving all the data points from which the model is to match as well as possible. The model output will be denoted as $y(k)$. It might be clearer to use some notation that distinguishes $y(k)$ as a model output variable and not the true system output variable. However, we will not have any need to write an expression for the true system, a system which we never know. Hence, for notational simplicity $y(k)$ is the model output.

Parameters: Each model contains certain parameters which must be chosen in order to make the model fit the data as well as possible. These parameters include the coefficients in the model, but they can also include the initial conditions for the model for the given data set. The parameters are collected as a column matrix ρ . When we wish to show explicitly the dependence on these parameters, the model output $y(k)$ will be written $y(k; \rho)$. The model output history and the measured history $y^*(k)$ from $k = k_o$ to $k = N$ are packaged as the column vector $\underline{y}(\rho)$ and \underline{y}^* .

Optimization Criterion: With these definitions, we can now conveniently write the objective function. We wish to minimize

$$J(\rho) = \frac{1}{2} [\underline{y}(\rho) - \underline{y}^*]^T [\underline{y}(\rho) - \underline{y}^*] \quad (1)$$

over the parameter set ρ . The objective is to find the model of the chosen type that minimizes the Euclidean norm of the model output error in fitting the data. This represents a nonlinear least squares problem, and we create a sequential quadratic programming approach to minimizing $J(\rho)$, making use of the fact that we can develop explicit expressions for both first and second derivative information for the change of J as a function of parameters ρ .

Derivative Notation: For notational simplicity, we indicate the derivative of a quantity with respect to some variable by using a subscript containing a comma followed by that variable. Hence, $J_{,\rho}(\rho)$ represents the column vector of the scalar J differentiated with respect to the elements of ρ in order from top to bottom. The derivative of a column vector with respect to a vector, as in $\underline{y}_{,\rho}(\rho)$, denotes the rectangular matrix with the first column being the derivative of $\underline{y}(\rho)$ with respect to the first element of ρ , the second column is the derivative of $\underline{y}(\rho)$ with respect to the second element of ρ , etc. The notation $y_{,\rho\rho}(k;\rho)$ indicates the Hessian matrix of second partial derivatives, with the i, j component being the second partial derivative with respect to the i th and the j th elements of ρ .

TAYLOR SERIES EXPANSION OF OPTIMIZATION CRITERION

The first and second derivatives of the objective function are written in terms of this notation as

$$J_{,\rho}(\rho) = [\underline{y}(\rho) - \underline{y}^*]^T \underline{y}_{,\rho}(\rho) \quad (2)$$

$$J_{,\rho\rho}(\rho) = \underline{y}_{,\rho}^T(\rho) \underline{y}_{,\rho}(\rho) + Q(\rho) \quad (3)$$

where

$$Q(\rho) = \sum_{k=k_0}^N [\underline{y}(k;\rho) - \underline{y}^*(k)]^T y_{,\rho\rho}(k;\rho)$$

Then, the quadratic approximation of the objective function, written in terms of parameter values ρ for the expansion point, and values $\rho + \Delta\rho$ at the evaluation point, becomes

$$\begin{aligned}
J(\rho + \Delta\rho) &\cong J(\rho) + J_{,\rho}(\rho)\Delta\rho + \frac{1}{2}\Delta\rho^T J_{,\rho\rho}(\rho)\Delta\rho \\
&\cong J(\rho) + [\underline{y}(\rho) - \underline{y}^*]^T \underline{y}_{,\rho}(\rho)\Delta\rho + \frac{1}{2}\Delta\rho^T \left[\underline{y}_{,\rho}^T(\rho)\underline{y}_{,\rho}(\rho) + Q(\rho) \right] \Delta\rho
\end{aligned} \tag{4}$$

THE NONLINEAR LEAST SQUARES ALGORITHM

Minimizing the Quadratic Expansion of the Cost: Sequential quadratic programming starts with an initial guess for the model parameters ρ , finds the change $\Delta\rho$ to minimize the quadratic approximation in (4), takes a step in direction $\Delta\rho$, and repeats the procedure until convergence.

The $\Delta\rho$ that minimizes this quadratic approximation satisfies

$$\left[\underline{y}_{,\rho}^T(\rho)\underline{y}_{,\rho}(\rho) + Q(\rho) \right] \Delta\rho = -\underline{y}_{,\rho}^T(\rho)[\underline{y}(\rho) - \underline{y}^*] \tag{5}$$

provided a finite minimum exists, i.e. provided $\left[\underline{y}_{,\rho}^T(\rho)\underline{y}_{,\rho}(\rho) + Q(\rho) \right]$ is at least positive semi-definite.

Choice of Updates in Indefinite Situations: The first term $\underline{y}_{,\rho}^T(\rho)\underline{y}_{,\rho}(\rho)$ can never have a negative eigenvalue, but it is possible for $Q(\rho)$ to have negative eigenvalues. In order to address the case where a finite minimum does not exist, we choose to perform an eigendecomposition on $Q(\rho)$, i.e. $Q(\rho) = M^{-1}\Lambda M$ where Λ is the diagonal matrix of eigenvalues. Then, we replace any negative eigenvalue in Λ by zero, and call the resulting matrix $\hat{\Lambda}$. Then let

$$\hat{Q}(\rho) = M^{-1}\hat{\Lambda}M \tag{6}$$

and use this in place of $Q(\rho)$ in the parameter update formula (5). Note that the sum of a positive definite matrix and a positive semi-definite matrix is positive definite. Therefore, the solution of (5) for $\Delta\rho$ exists.

The Update Law and Step-Size Control: The general nonlinear least-squares algorithm to be applied to each of the five model types, can now be stated. Starting from some initial parameter estimates ρ_0 , we iteratively update the parameters according to

$$\rho_{j+1} = \rho_j - a_i \left[\underline{y}_{,\rho}^T(\rho_j) \underline{y}_{,\rho}(\rho_j) + \hat{Q}(\rho_j) \right]^{-1} \underline{y}_{,\rho}^T(\rho_j) [\underline{y}(\rho_j) - \underline{y}^*] \quad i = 0, 1, 2, \dots \quad (7)$$

The coefficient a_i is a step size which in the numerical results reported below is adjusted as follows. Each iteration starts with $a_i = 1$, and the result is checked to see whether the resulting system is asymptotically stable, and to see if the associated cost (1) is decreased. If both conditions are satisfied, then the step is taken. Otherwise, the value of a_i is successively decreased to 0.9, 0.8, ... , 0.1, 0.09, ... , 0.01, 0.009, ... , 0.001, etc. until both conditions are met. The user specifies a condition for termination of the iterations. If the Euclidean norm of the error between the model output and the data at iteration $i+1$, minus the same quantity at i , divided by the norm of the error at iteration i , is less than a pre specified value, the iteration is stopped.

Other Possible Algorithm Choices: Note that if the eigendecomposition is considered too time consuming, one could adopt various solvers that are designed to handle indefinite Hessians. It is also possible to avoid the eigendecomposition by using a modified Cholesky decomposition for the whole exact Hessian matrix. In the examples discussed below, the eigendecomposition was quite tractable, with quite reasonable run times, and it is of course the best, or most robust way to handle the non-definite second partial problem. A good text discussing the range of possible methods is [14].

What Remains to be Specified to Complete the Algorithm for each Model Type: In order to complete the system identification algorithm, two things remain:

1. One is the choice of method to obtain the starting values ρ_0 , discussed in the next section.
2. Second is the development of explicit methods to evaluate the derivatives $\underline{y}_{,\rho}(\rho)$ and $y_{,\rho\rho}(k; \rho)$ for the five choices of model types. These are discussed successively in the sections that follow.

OKID INITIALIZATION OF NONLINEAR LEAST SQUARES ALGORITHM

In this section, we review the OKID algorithm and discuss what criterion it optimizes in producing its identification results. It is seen that in limiting cases this criterion approaches that being used here, so that employing OKID to get the starting values is a very good choice -- the OKID results should be close to the optimum we are searching for in the nonlinear least-squares iterations.

Suppose that the data corresponds to a linear system of the form

$$\begin{aligned}x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k)\end{aligned}\tag{8}$$

Then a linear observer takes the form

$$\begin{aligned}\hat{x}(k+1) &= A\hat{x}(k) + Bu(k) - G[y(k) - \hat{y}(k)] = \bar{A}\hat{x}(k) + \bar{B}v(k) \\ \hat{y}(k) &= C\hat{x}(k) + Du(k)\end{aligned}\tag{9}$$

where the right hand side of the difference equation contains the measurement $y(k)$ with the observer gain M . The coefficient matrices are related by $\bar{A} = A + GC$, $\bar{B} = [B + GD, -G]$, $v(k) = [u^T(k) \quad y^T(k)]^T$.

There is a convolution sum version of the observer (9)

$$y(k) = Du(k) + C\bar{B}v(k-1) + \dots + C\bar{A}^{p-1}\bar{B}v(k-p) + C\bar{A}^p x(k-p)\tag{10}$$

where p is a time-step integer. Equation (10) can be packaged for the data set being used as follows

$$\bar{y} = \bar{Y}\bar{V} + \varepsilon + C\bar{A}^p X\tag{11}$$

where

$$\begin{aligned}
\bar{y} &= [y(p+1) \quad y(p+2) \quad \cdots \quad y(\ell)]; \\
\varepsilon &= [\varepsilon_r(p+1) \quad \varepsilon_r(p+2) \quad \cdots \quad \varepsilon_r(\ell)]; \\
X &= [x(1) \quad x(2) \quad \cdots \quad x(\ell-p-1)]; \\
\bar{Y} &= [D \quad C\bar{A} \quad C\bar{A}^2 \quad \cdots \quad C\bar{A}^{p-1}\bar{B}]; \\
\bar{V} &= \begin{bmatrix} u(p+1) & u(p+2) & \cdots & u(\ell) \\ v(p) & v(p+1) & \cdots & v(\ell-1) \\ v(p-1) & v(p) & \cdots & v(\ell-2) \\ \vdots & \vdots & \ddots & \vdots \\ v(1) & v(2) & \cdots & v(\ell-p) \end{bmatrix}.
\end{aligned}$$

The quantity ε is introduced to represent equation error when the data in \bar{y} and \bar{V} are inconsistent due to noise.

The matrix \bar{A} is an observer system matrix. In the noise free case it can be made deadbeat, and in the presence of noise, it can correspond to a Kalman filter. In either case it is asymptotically stable, and there will be a value of p for which \bar{A}^p is negligible. In using OKID one picks such a p making the last term of (11) negligible, and performs a least squares solution to the remainder of the equation, using

$$\bar{Y} = \bar{y}\bar{V}^T [\bar{V}\bar{V}^T]^{-1} \quad (12)$$

which minimizes the error measure $\varepsilon^T \varepsilon$, where ε is the difference between the \bar{y} obtained using this model (with \bar{A}^p neglected) and measured values. Then from \bar{Y} the Markov parameters of the system are computed. From these the modern control realization of the form of Eq. (8) is obtained by the ERA-DC algorithm [12]. This is a nonlinear computation and noise in the data can therefore introduce biases [16], but as noted below biases disappear under appropriate conditions.

We can make the following conclusions:

1. OKID minimizes the optimization function of interest, the error between output of the model (10) and measured output, but this is not done for a model (8) of the system itself. Instead it is for an observer of the system. This differs from the optimization we wish to perform here because the previous outputs appear on the right hand side of the equation (10).
2. If the main point of the identification is to obtain an observer, for example for use in feedback control, then OKID is directly addressing the appropriate objective function.
3. The observer is in the convolution sum form (10), and neglects terms containing \bar{A}^p . Before an observer is used, it will most likely be put in a recursive state variable difference equation form (9). In this case, it is possible that using the additional nonlinear least-squares optimization on the state variable model coefficients could produce improved results. Getting from the observer Markov parameters, i.e., coefficient matrices in (10), to the minimal state variable model (9) involves nonlinear equations. Thus the optimization in state space form (8) can produce a somewhat different result than in convolution sum form (10). Note that the assumption of \bar{A}^p being negligible is not used in the process of the optimization.

We have stated that under appropriate conditions, the observer identified by OKID converges to the steady state Kalman filter. The development of a Kalman filter assumes that the system behavior can be described by a model of the form

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + r(k) \\ y(k) &= Cx(k) + Du(k) + s(k) \end{aligned} \tag{13}$$

where $r(k)$ and $s(k)$ are zero mean, white, and Gaussianly distributed noise. Post multiply Eq. (11) by \bar{V}^T , to obtain

$$\bar{y}\bar{V}^T = \bar{Y}(\bar{V}\bar{V}^T) + \varepsilon\bar{V}^T + C\bar{A}^p X\bar{V}^T \tag{14}$$

The choice of the identified observer model in (12) makes the left hand side and the first term on the right match in this equation. Examine the second term on the right. The component of $\varepsilon\bar{V}^T$ formed by the product of ε with the row of \bar{V} that is $i+1$ from the bottom can be written as

$$\sum_{j=1}^{\ell-p} \varepsilon_r(p+j)v^T(i+j-1); \quad i=1,\dots,p \quad (15)$$

If this term is divided by $\ell-p$, it represents the time average of the product $\varepsilon(k)v^T(k-p-1+i)$ from $k = p+1$ to ℓ . By the ergodic property, if the product is a sample function of a stationary random process, it can be replaced by its ensemble average provided that ℓ goes to infinity, $\ell \rightarrow \infty$

$$E[\varepsilon_r(k)v^T(k-p-1+i)] = \lim_{\ell \rightarrow \infty} \frac{1}{\ell-p} \sum_{j=p+1}^{\ell} \varepsilon_r(j)v^T(j-p-1+i); \quad k > p \quad (16)$$

Then the terms remaining in Eq. (14) become

$$\begin{aligned} & E[\varepsilon_r(k)v^T(k-1) \quad \varepsilon_r(k)v^T(k-2) \quad \dots \quad \varepsilon_r(k)v^T(k-p)] \\ & = -C\tilde{A}^p E[X(k)v^T(k+p-1) \quad X(k)v^T(k+p-2) \quad \dots \quad X(k)v^T(k)] \end{aligned} \quad (17)$$

Using the assumption that \bar{A}^p is negligible, we conclude that the procedure indicates $E[\varepsilon_r(k)v^T(k-i)] = 0$ for $i = 1, 2, \dots, p$.

We can now make a second set of conclusions:

1. Suppose that the real world is governed by a model of the form of Eq. (13), a finite dimensional linear model with additive zero mean white Gaussian plant and measurement noise.
2. Suppose that \bar{A}^p is negligible.
3. Suppose that the data set is sufficiently long that one can use the ergodic property to replace a time average by an expected value.
4. Then the quantity $\varepsilon(k)$ is the innovations term, the difference between the measured output and the output predicted by the observer, which is white in a Kalman filter.
5. The observer satisfies the equation $E[\varepsilon(k)y^T(k-i)] = 0$; $i = 1, \dots, k$ which is the orthogonality principle. When an observer satisfies this orthogonality principle, it is a Kalman filter [15], and the estimated output is unbiased.

6. A Kalman filter has the property that it minimizes the variance of the (white) error in the estimated output for each time step.
7. This is equivalent to minimizing the sum of the variances (or traces of the covariances in the multiple output case) of these white errors. And this can be written as the time average in place of the expectation.
8. This time average is the optimization criterion we wish to minimize in this paper.
9. The OKID produces a result that satisfies the optimality condition proposed in this paper, when the true world is governed by a model with additive plant and measurement noise, \bar{A}^p is negligible, and a sufficiently long data set is used such that time average to coincide with the ensemble average.
10. The fact that the OKID result satisfies the optimality condition we wish to satisfy in the limiting situation described here, means that it is a particularly good choice as starting conditions for the nonlinear least squares algorithm in this paper.
11. In so far as these assumptions are not met, the algorithms developed in this paper can be used to improve the OKID result.
12. When the data set is short, violating the long data record assumption here, the OKID with residual whitening algorithm [7,8] can be of help to produce starting values that are closer to the optimal solution we are looking for, by more closely satisfying the orthogonality principle. Also, the algorithms in [9,10] can have improved satisfaction of the orthogonality property in short data records. Here we restrict ourselves to use of OKID itself.

IDENTIFICATION OF MISO SYSTEMS USING AN ARX REALIZATION

In this section we consider multiple-input, single-output ARX models (MISO) of the form

$$\begin{aligned}
 y(k) = & \alpha_1 y(k-1) + \alpha_2 y(k-2) + \dots + \alpha_p y(k-p) \\
 & + \beta_0 u(k) + \beta_1 u(k-1) + \dots + \beta_p u(k-p)
 \end{aligned} \tag{18}$$

and develop the first and second partial derivatives of the output as a function of the model parameters, $y_{,\rho}(k; \rho)$ and $y_{,\rho\rho}(k; \rho)$, needed in the nonlinear least squares algorithm outlined previously. For simplicity of notation we will often drop the parameter argument ρ in the notation $y(k)$. The parameters of the model are the scalars $\alpha_1, \alpha_2, \dots, \alpha_p$ and the row vectors

$\beta_0, \beta_1, \dots, \beta_p$. A method to include the initial conditions into the parameter list will be discussed later. Now, let the requisite number of values of $y(k)$ and $u(k)$ from the data be used as initial conditions for Eq. (18), i.e.

$$y(1), y(2), \dots, y(p) \quad ; \quad u(1), u(2), \dots, u(p) \quad (19)$$

when we assume a model of order p and allow a β_0 term (for example, when using collocated accelerometer outputs in structural vibrations such a term is needed). This approach is useful when there is no noise in its initial information, or the inaccuracies of these initial values, which decay over a time interval of roughly one settling time, makes negligible error when the full data set is considered.

Now consider the evaluation of the components of $\underline{y}_{,\rho}(\rho)$ that are derivatives with respect to the parameter α_i in ρ . Differentiation of Eq. (18) with respect to $\alpha_1, \alpha_2, \dots, \alpha_p$ produces the following sensitivity equations that couple in one direction to (18)

$$\begin{aligned} y_{,\alpha_1}(k) &= \alpha_1 y_{,\alpha_1}(k-1) + \alpha_2 y_{,\alpha_1}(k-2) + \dots + \alpha_n y_{,\alpha_1}(k-p) + y(k-1) \\ y_{,\alpha_2}(k) &= \alpha_1 y_{,\alpha_2}(k-1) + \alpha_2 y_{,\alpha_2}(k-2) + \dots + \alpha_n y_{,\alpha_2}(k-p) + y(k-2) \\ y_{,\alpha_3}(k) &= \alpha_1 y_{,\alpha_3}(k-1) + \alpha_2 y_{,\alpha_3}(k-2) + \dots + \alpha_n y_{,\alpha_3}(k-p) + y(k-3) \\ &\vdots \end{aligned} \quad (20)$$

Since the initial conditions in (19) in the present formulation come from data, their sensitivity to parameters of the model must be zero. This means that the initial conditions for the first equation in (20) are $y_{,\alpha_i}(1) = y_{,\alpha_i}(2) = \dots = y_{,\alpha_i}(p) = 0$ for all i . Note that the forcing function for the $y_{,\alpha_{i+1}}(k)$ sensitivity equation is the same as that for the $y_{,\alpha_i}(k-1)$ sensitivity equation except that it is delayed by one time step.

The sensitivities of $y(k)$ with respect to the i th component of the row vectors $\beta_0, \beta_1, \dots, \beta_p$ behave similarly, satisfying

$$\begin{aligned}
y_{,\beta_{0i}}(k) &= \alpha_1 y_{,\beta_{0i}}(k-1) + \alpha_2 y_{,\beta_{0i}}(k-2) + \dots + \alpha_n y_{,\beta_{0i}}(k-p) + u_i(k) \\
y_{,\beta_{1i}}(k) &= \alpha_1 y_{,\beta_{1i}}(k-1) + \alpha_2 y_{,\beta_{1i}}(k-2) + \dots + \alpha_n y_{,\beta_{1i}}(k-p) + u_i(k-1) \\
y_{,\beta_{2i}}(k) &= \alpha_1 y_{,\beta_{2i}}(k-1) + \alpha_2 y_{,\beta_{2i}}(k-2) + \dots + \alpha_n y_{,\beta_{2i}}(k-p) + u_i(k-2) \\
&\vdots
\end{aligned} \tag{21}$$

with the same statements applying to the initial conditions, and the same time shift condition applying to the forcing functions.

The second partial derivatives satisfy

$$y_{,\alpha_i \alpha_j}(k) = \alpha_1 y_{,\alpha_i \alpha_j}(k-1) + \alpha_2 y_{,\alpha_i \alpha_j}(k-2) + \dots + \alpha_p y_{,\alpha_i \alpha_j}(k-p) + y_{,\alpha_i}(k-j) + y_{,\alpha_j}(k-i) \tag{22}$$

$$y_{,\alpha_i \beta_{ji}}(k) = \alpha_1 y_{,\alpha_i \beta_{ji}}(k-1) + \alpha_2 y_{,\alpha_i \beta_{ji}}(k-2) + \dots + \alpha_p y_{,\alpha_i \beta_{ji}}(k-p) + y_{,\beta_{ji}}(k-i) \tag{23}$$

Again, the initial conditions on $y(k)$ are independent of the model, and hence their sensitivities with respect to the model parameters are zero, which gives zero initial conditions for these equations. Note that the sensitivity equations for $y_{,\beta_{ji} \beta_{jm}}(k)$ are homogeneous equations with zero initial conditions, and therefore these sensitivities are zero for all time steps.

From equations (20) and (21), it is noted that the forcing functions were all the same except for a time shift. This suggests that the solutions might also be the same except for a time shift. If this is true, then the equations for the second partials in (22) and (23) are fed by forcing functions that again are all the same except for a time shift and their solutions can again be all the same with time shifts. When this property applies to these sensitivity equations, it is possible to obtain a very substantial compression in computation requirements, because one needs to solve only one equation for each type of sensitivity, and then obtain all the rest by shifting the time argument an appropriate number of steps.

The solutions to the sensitivity (20) and (21) will all be pure time shifts of each other provided it is possible to set up the experiments so that the initial conditions on $y(k)$ and $u(k)$ in Eq. (19) are zero. Even when one cannot arrange that this applies to the data at hand, one may still be able to take advantage of this compression in computation time, and simply apply time shifted results

in the computations. The differences in solutions at the start will likely be small, and will disappear in roughly one settling time of the system, and hence they may have insignificant effect on the convergence of the algorithm.

When taking advantage of these shifting properties of the solutions, one solves the following set of difference equations

$$\begin{aligned}
y(k) &= \alpha_1 y(k-1) + \alpha_2 y(k-2) + \dots + \alpha_p y(k-p) + \beta_0 u(k) + \beta_1 u(k-1) + \dots + \beta_p u(k-p) \\
y_{,\alpha_1}(k) &= \alpha_1 y_{,\alpha_1}(k-1) + \alpha_2 y_{,\alpha_1}(k-2) + \dots + \alpha_p y_{,\alpha_1}(k-p) + y(k-1) \\
y_{,\beta_{0i}}(k) &= \alpha_1 y_{,\beta_{0i}}(k-1) + \alpha_2 y_{,\beta_{0i}}(k-2) + \dots + \alpha_p y_{,\beta_{0i}}(k-p) + u_i(k) \\
y_{,\alpha_1 \alpha_1}(k) &= \alpha_1 y_{,\alpha_1 \alpha_1}(k-1) + \alpha_2 y_{,\alpha_1 \alpha_1}(k-2) + \dots + \alpha_p y_{,\alpha_1 \alpha_1}(k-p) + 2y_{,\alpha_1}(k-1) \\
y_{,\alpha_1 \beta_{0i}}(k) &= \alpha_1 y_{,\alpha_1 \beta_{0i}}(k-1) + \alpha_2 y_{,\alpha_1 \beta_{0i}}(k-2) + \dots + \alpha_p y_{,\alpha_1 \beta_{0i}}(k-p) + y_{,\beta_{0i}}(k-1)
\end{aligned} \tag{24}$$

Then the remaining sensitivities are obtained by the following shifting equations

$$\begin{aligned}
y_{,\alpha_i}(k) &= y_{,\alpha_{i-1}}(k-1) = \dots = y_{,\alpha_1}(k-i+1) \\
y_{,\beta_{i\ell}}(k) &= y_{,\beta_{i-1,\ell}}(k-1) = \dots = y_{,\beta_{0\ell}}(k-i) \\
y_{,\alpha_i \alpha_j}(k) &= y_{,\alpha_1 \alpha_1}(k-i-j+2) \\
y_{,\alpha_i \beta_{j\ell}}(k) &= y_{,\alpha_1 \beta_{0\ell}}(k-i-j+1)
\end{aligned} \tag{25}$$

together with the fact that all $y_{,\beta_i \beta_{jm}}(k)$ are zero. Examples of use of this algorithm are given below.

IDENTIFICATION OF MISO SYSTEMS USING THE OBSERVABLE STATE SPACE REALIZATION

In this section we again address MISO systems, but do so in a modern state variable formulation, using what might be called the left observable canonical form. We will simply call it observable form. The model is written as

$$\begin{aligned}
x(k+1) &= Ax(k) + Bu(k) \\
y(k) &= Cx(k) + Du(k)
\end{aligned} \tag{26}$$

where

$$A = \begin{bmatrix} \alpha_1 & 1 & 0 & \cdots & 0 \\ \alpha_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_{n-1} & 0 & 0 & \cdots & 1 \\ \alpha_p & 0 & 0 & \cdots & 0 \end{bmatrix} \quad B = \begin{bmatrix} \beta_1 + \alpha_1 \beta_0 \\ \beta_2 + \alpha_2 \beta_0 \\ \vdots \\ \beta_p + \alpha_p \beta_0 \end{bmatrix} \quad C = [1 \quad 0 \quad \cdots \quad 0] \quad D = \beta_0$$

This model is equivalent to model (18) used in the previous section. Identification results were identical using two programs, one for each formulation. Nevertheless, certain things are much easier to see using one formulation than the other, and in addition, this formulation is more straightforward to code.

The first and second partial derivatives satisfy the sensitivity equations

$$\begin{aligned} x_{,\rho}(k+1) &= Ax_{,\rho}(k) + A_{,\rho}x(k) + B_{,\rho}u(k) \\ y_{,\rho}(k) &= Cx_{,\rho}(k) + D_{,\rho}u(k) \\ x_{,\rho_i\rho_j}(k+1) &= Ax_{,\rho_i\rho_j}(k) + A_{,\rho_i}x_{,\rho_j}(k) + A_{,\rho_j}x_{,\rho_i}(k) + B_{,\rho_i\rho_j}u(k) \\ y_{,\rho_i\rho_j}(k) &= Cx_{,\rho_i\rho_j}(k) \end{aligned} \tag{27}$$

The matrix derivatives involved are very easy to calculate:

$A_{,\alpha_i}$ is all zero except for a one in the i th row of the first column;

$A_{,\beta_{ij}}$ is a zero matrix;

$B_{,\alpha_i}$ is all zero except for β_0 in the i th row;

$B_{,\beta_{ij}}$ is all zero except for a one in the i th row and j th column; all derivatives in

$D_{,\rho}$ is a zero matrix except for the $D_{,\beta_{0i}}$ which is a zero row except for a one in the i th column;

$B_{,p_i p_j}$ is all zero except those that involve the combinations of α_i and β_{0j} which are zero matrices except for a one in the i th row and j th column.

$A_{,p_i p_j}$ is a zero matrix for any combination of parameters;

$B_{,p_i p_i}$ is a zero matrix.

The initial conditions are $x(0) = x_o$, $x_{,p}(0) = 0$, and $x_{,p_i p_j}(0) = 0$. Using this form of the sensitivity equations, the programming of the computations becomes very straightforward.

It is interesting to package the equations in terms of the state equation

$$\begin{aligned} & \begin{bmatrix} x_{,\alpha_1}(k+1) & \cdots & x_{,\alpha_p}(k+1) & x_{,\beta_0}(k+1) & \cdots & x_{,\beta_p}(k+1) \end{bmatrix} \\ &= A \begin{bmatrix} x_{,\alpha_1}(k) & \cdots & x_{,\alpha_p}(k) & x_{,\beta_0}(k) & \cdots & x_{,\beta_p}(k) \end{bmatrix} + F \end{aligned}$$

and output equation

$$\begin{aligned} & \begin{bmatrix} y_{,\alpha_1}(k) & \cdots & y_{,\alpha_p}(k) & y_{,\beta_0}(k) & y_{,\beta_1}(k) & \cdots & y_{,\beta_p}(k) \end{bmatrix} \\ &= C \begin{bmatrix} x_{,\alpha_1}(k) & \cdots & x_{,\alpha_p}(k) & x_{,\beta_0}(k) & x_{,\beta_1}(k) & \cdots & x_{,\beta_p}(k) \end{bmatrix} \\ &+ \begin{bmatrix} 0 & \cdots & 0 & u(k) & 0 & \cdots & 0 \end{bmatrix} \end{aligned}$$

where

$$F = \begin{bmatrix} x_1(k) + Du(k) & 0 & \cdots & 0 & \alpha_1 u^T(k) & u^T(k) & 0 & \cdots & 0 \\ 0 & x_1(k) + Du(k) & \cdots & 0 & \alpha_2 u^T(k) & 0 & u^T(k) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & x_1(k) + Du(k) & \alpha_p u^T(k) & 0 & 0 & \cdots & u^T(k) \end{bmatrix}$$

As mentioned before, the algorithm of this section must produce identical results as that of the previous section, but is more easily programmed in a language such as MATLAB. Since matrix manipulations in that language are efficient, there is little incentive to make a more specialized program to take advantage of the number of zeros present in the matrices. Which approach is better computationally is perhaps language dependent and also dependent on the programming details. The observable canonical form here makes the β_0 term appear fundamentally different,

whereas in the ARX version it appears like just another β_i . The shifting properties of the sensitivities are much more easily viewed by the ARX formulation. Both approaches were programmed with identical results.

IDENTIFICATION OF THE INITIAL CONDITION

The initial conditions for the model are the first p steps of the output data set. So far, we have treated these measured values as fixed numbers that are unrelated to the coefficients in the model. However, since they are measured values, they can contain noise, and one of the functions of the model is to smooth through the noise. Of course, the influence of small changes in initial values will disappear in the output in one settling time of the system, and hence are likely not to be important in most applications. If the system is very lightly damped or the data record short, one might want to let the optimization criterion apply to the initial conditions as well, to smooth through the noise in the initial p data points. Hence we include the initial conditions among the parameter vector ρ that are used in the optimization. This means that we need to develop one more set of sensitivity equations for these new parameters. The first p data points used to serve as initial conditions are relegated now to simply starting conditions on the nonlinear least squares iteration.

The system matrices are not functions of the initial condition, so their first and second partials with these new parameters are zero. Coupling with the β_i is zero. In addition, second derivatives with respect to initial conditions are zero. The only sensitivity equations needed are

$$\begin{aligned} x_{,x_i(0)}(k+1) &= Ax_{,x_i(0)}(k) \\ x_{,\alpha_i x_j(0)}(k+1) &= Ax_{,\alpha_i x_j(0)}(k) + A_{,\alpha_i} x_{,x_j(0)}(k) \end{aligned} \tag{28}$$

with initial conditions $x_{,x_i(0)}(0)$ equal to a column matrix of zeros except for a one in the i th entry, and $x_{,\alpha_i x_j(0)}(0)$ equal to zero.

IDENTIFICATION OF MIMO ARX MODELS

The ARX model of Eq. (18) was for a multiple-input, single-output model which makes the notation more straightforward. To handle multiple-input, multiple-output models (MIMO) there is a range of choices of generalized ARX models that one can use. With reference to the ARX model of Eq. (18), the output $y(k)$ must become column vectors of the multiple outputs at each time step. Denote the minimal order of the MIMO system as n , the number of the square α_i matrices as p , the number of inputs as r , and the number of outputs as m .

One of the possible ARX formulation maintains the α_i as scalars. In this case they are the coefficients of the MIMO characteristic equation, and there are $p = n$ of them. There are $n + 1$ matrices β_i of dimension m by r . The number of unknown coefficients in the model that must be identified is $n + (n + 1)mr$.

Another ARX formulation allows the α_i to be $m \times m$ matrices. Then the number m of the α_i used must satisfy the constraint that $m \times p \geq n$ (this becomes obvious when the MIMO equation is put in observable form as in the next section). Then the number of unknowns in the α_i is pm^2 . The number of $m \times r$ matrices β_i becomes $p + 1$, and the number of unknowns in them becomes $(p + 1)mr$. So the number of coefficients in the model becomes $pm^2 + (p + 1)mr$. If the integer p can be chosen such that $m \times p = n$, then $pm^2 + (p + 1)mr = n(m + r) + mr$.

Different values of m can have a significant effect on the number of coefficients that must be determined. For the first case the number of coefficients in α_i is n , which is to be compared with pm^2 in the second case. Note that pm is greater than n , so that the number of such coefficients is increased when one goes to matrix α_i . On the other hand the number of β_i matrices decreases from $n + 1$ to $p + 1$, and the number of unknowns involved decreases by $(n - p)mr$.

For any choice of m , the only changes needed to create an identification algorithm for the ARX model generalized to allow square matrix α_i and rectangular matrix β_i is to rewrite the sensitivity equations in (18) to (25) to be sensitivities with respect to components of each matrix.

For example, writing the i, j component of the square coefficient matrix α_ℓ as $\alpha_{\ell ij}$, the sensitivity equations corresponding to Eq. (20) are

$$y_{,\alpha_{ij}}(k) = \alpha_1 y_{,\alpha_{ij}}(k-1) + \alpha_2 y_{,\alpha_{ij}}(k-2) + \dots + \alpha_p y_{,\alpha_{ij}}(k-p) + \alpha_{\ell,\alpha_{ij}} y(k-\ell) \quad (29)$$

where $\alpha_{\ell,\alpha_{ij}}$ is all zero except for a one in the i th row and j th column. The other equations are similar.

IDENTIFICATION OF MIMO MODELS IN OBSERVABLE FORM

We can also generalize the algorithm for observable canonical form quite simply. The system matrices shown in Eq. (26) become

$$A = \begin{bmatrix} \alpha_1 & I & 0 & \dots & 0 \\ \alpha_2 & 0 & I & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_{m-1} & 0 & 0 & \dots & I \\ \alpha_m & 0 & 0 & \dots & 0 \end{bmatrix} ; \quad C = [I \quad 0 \quad \dots \quad 0] \quad (30)$$

where I is an identity matrix of dimension $m \times m$. If the choice of p is such that $pm = n$, then this is a minimal realization, and when $pm > n$ it is not. Since α_i and β_i are square and rectangular matrices, we need to take sensitivities with respect to each component. For example,

$$A_{,\alpha_{1ij}} = \begin{bmatrix} \alpha_{1,\alpha_{1ij}} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} \quad (31)$$

The shifting properties shown in Eq. (25) still apply for each component.

IDENTIFICATION OF MIMO MODELS IN BLOCK DIAGONAL FORM

In modal testing and in control of vibrations in structures such as spacecraft, one knows that the model should be composed of a set of vibration modes. This corresponds to having all eigenvalues of the system matrix be in complex conjugate pairs. Let us assume that the system matrix is diagonalizable, as is normally the case in structural dynamic systems even with repeated eigenvalue pairs. In these applications, one is particularly interested in knowing the frequencies of the modes and their damping ratios. One can of course use any of the above methods for this class of problems, and then find the characteristic polynomial and its roots, or equivalently calculate the eigenvalues of the state matrix. Here we develop a method that finds these quantities directly. The method can be generalized to include real roots, but to do so one must address issues of knowing how many real roots to allow, or how to facilitate conversion from real to complex and vice versa. One also must address such issues as when the starting values for the nonlinear least squares fail to satisfy the conjugate pair requirement.

Because the roots are in complex conjugate pairs, there is a realization in block diagonal form. This means that the output can be written as a superposition of the outputs of n_2 decoupled systems (overall system order is $n = 2n_2$),

$$x_j(k+1) = A_j x_j(k) + B_j u(k) \quad j = 1, 2, \dots, n_2 \quad (32)$$

$$y(k) = Cx(k) + Du \quad (33)$$

one for each complex conjugate eigenvalue pair $\lambda_j \pm i\omega_j$, where there are r inputs in u and m outputs in y , and

$$\begin{aligned} x_j(k) &= \begin{bmatrix} x_{j1}(k) \\ x_{j2}(k) \end{bmatrix}; \\ A_j &= \begin{bmatrix} \lambda_j & \omega_j \\ -\omega_j & \lambda_j \end{bmatrix} ; \\ B_j &= \begin{bmatrix} b_{j11} & b_{j12} & \cdots & b_{j1r} \\ b_{j21} & b_{j22} & \cdots & b_{j2r} \end{bmatrix} \end{aligned}$$

and

$$x(k) = \begin{bmatrix} x_{11}(k) & x_{12}(k) & \cdots & x_{n_2 1}(k) & x_{n_2 2}(k) \end{bmatrix}^T$$

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} & c_{1n} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} & c_{mn} \end{bmatrix}$$

$$D = \begin{bmatrix} d_{11} & \cdots & d_{1r} \\ \vdots & \cdots & \vdots \\ d_{m1} & \cdots & d_{mr} \end{bmatrix}$$

The nonlinear least squares optimization with respect to the parameters in the matrices A_j and B_j is performed as shown earlier after developing the appropriate sensitivity equations. It can be performed on the matrix level as in the observable canonical form algorithm. The optimization with respect to C and D can be done in a more direct way. Note that

$$y_{.c_{ij}}(k) = C_{.c_{ij}} x(k) \quad ; \quad y_{.d_{ij}}(k) = D_{.d_{ij}} u(k)$$

where the coefficient matrices are all zero except for one entry that is unity, and the second partials are zero. Here we let the nonlinear least squares update the parameters in A_j and B_j , and then for the associated state history we find the pseudo inverse solution for estimated C and D from the observation equations (33) packaged together for all time steps $k = 1, 2, \dots, \ell$ in the data

$$\begin{bmatrix} y(1) & \cdots & y(\ell) \end{bmatrix} = \begin{bmatrix} C & D \end{bmatrix} \begin{bmatrix} x(1) & \cdots & x(\ell) \\ u(1) & \cdots & u(\ell) \end{bmatrix} \quad (34)$$

Such a pseudo inverse finds the C and D matrices that minimize the cost (or sum of the squares of the output errors) for given A_j and B_j .

Besides having the advantage of directly identifying the quantities of most interest, i.e., the frequencies and damping factors for each mode, this approach can have a dimensionality advantage. The number of coefficients in this model that must be chosen by optimization is $n + nr + nm + mr$ which is the sum of the following: n scalars for eigenvalues, nr for matrix B ,

nm for C , and mr for D . Comparing it to the MIMO ARX model with scalar α_i , this realization has fewer parameters if $r + m < rm$.

NUMERICAL EXAMPLE

In this section we present some numerical examples to show the effectiveness of the algorithms developed. First we consider the three mass system shown in Fig. 1:

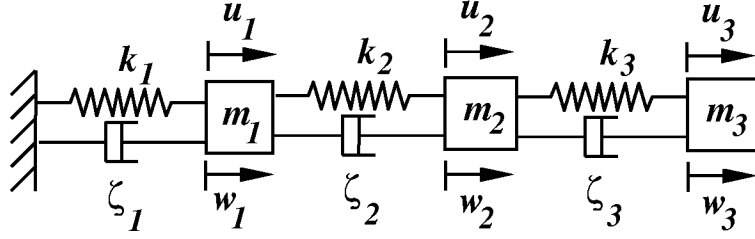


Figure 1: Three mass-spring-dashpot system

whose equations of motion can be written as

$$M\ddot{w} + \Xi\dot{w} + Kw = Bu$$

$$y = C_a\ddot{w} + v$$

where

$$w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix};$$

$$M = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix}$$

$$\Xi = \begin{bmatrix} \zeta_1 + \zeta_2 & -\zeta_2 & 0 \\ -\zeta_2 & \zeta_2 + \zeta_3 & -\zeta_3 \\ 0 & -\zeta_3 & \zeta_3 \end{bmatrix}$$

$$K = \begin{bmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{bmatrix}$$

The three-dimensional column vector w is the inertial displacement of the three masses starting at the left for the top entry. We make a single-input, single-output system with the input u being a force applied to the right mass, and the output is an inertial acceleration measurement of the left mass. This means that column matrix B is zero except for one in the third entry, and row vector C_a is unity in its left entry followed by two zeros. The mass matrix is taken as the identity matrix, each of the k_i in the stiffness matrix K are set to 1000, and the damping matrix Ξ has the same form as K with the k_i replaced by 0.1. The v represents white zero-mean Gaussian measurement noise with variance equal to unity. Simulated experimental data is generated numerically solving these equations to produce 1000 data points with a sample time of 0.05 seconds. The input $u = u_3$ is a white zero-mean Gaussian input also of variance equal to unity. Thus the measurement noise level in this data is quite high. The ARX algorithm developed above for MISO systems is used to produce an identification. In addition, a different program for the observable state space realization is applied, producing identical results.

Table 1 gives the true values of the frequencies and the dampings for each mode. The frequencies are the undamped natural frequencies in Hertz (given by the magnitude of the system eigenvalues divided by 2π), and the damping in percent is given by 100 times the real part of the root divided by its magnitude. Note that the system is very lightly damped.

OKID is used on the data set to obtain the initial identification. This algorithm has a parameter p which is chosen as 6 in this run. The results are given in Table 1, and we see that the third frequency is not very accurate, and that the dampings are all over estimated, sometimes by a very large margin. This over estimation of damping is a very common phenomenon in modal identification with algorithms such as OKID and ERA, ERA-DC.

Table 1. Comparison of OKID-identified and ARX-optimized frequencies and dampings.

Mode No.	True Values		OKID Result		Optimized Identification	
	Freq (Hz)	Damping %	Freq (Hz)	Damping %	Freq (Hz)	Damping %
1	2.240	0.070	2.215	2.537	2.240	0.080
2	6.276	0.197	6.274	0.361	6.276	0.196
3	9.069	0.285	9.873	12.472	9.069	0.270

We now apply the MISO ARX algorithm developed here to the data set, starting with the OKID result. Instead of quoting the values of the optimization criterion (1) we take the square root of it to produce the norm of the error. The initial value from the OKID result is 130.974. After 19 iterations this norm has stabilized to 6 digits and the iterations are terminated. The final norm of the error is 31.2948. Note that the norm of the measurement noise alone is 31.430. The fact that this number is slightly larger than the final error level means that a small part of the noise has been fit in the model. It would be impossible to meaningfully get to a lower error level with further iterations, or by using any other algorithm.

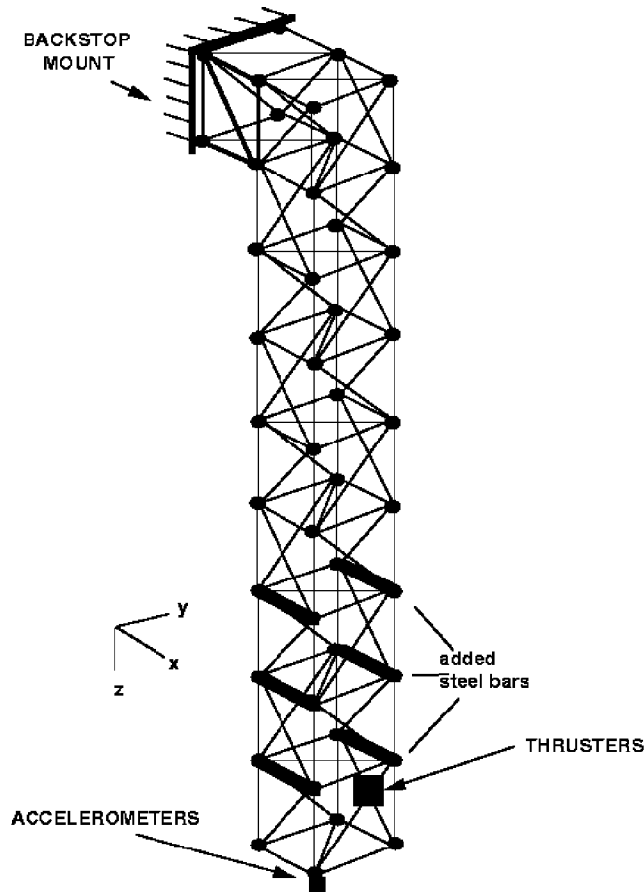
Examination of the identified frequencies and dampings shows that in spite of the very high noise level, the results are quite good. The damping for each mode is close to the true value, so that the optimized identification method developed here is seen to fix the persistent problem of overestimating the damping in modal identification.

The first iteration used a step size of 0.5 and decreased the norm of the error to 120. The second iteration used a step size of 1 and reached an error norm of 80. By iteration 6 the error was 56. The step size was unity for all but 5 iterations with the nonunity values ranging from 0.2 to 0.7. One issue of importance in this class of problems is the run times to get results. When a standard nonlinear least-squares algorithm was used without having developed the derivative information as in this paper, the run times are prohibitive. The run time for this optimized identification took 0.81823 minutes on a 200 megaHertz pentium PC using Matlab version 5.0. Thus, the computation times are reasonable, making the algorithm practical for real engineering applications.

EXPERIMENTAL EXAMPLE

The experimental results given in this section illustrate the usefulness of the proposed optimization technique in practice. The truss structure at NASA Langley Research Center is shown in Fig. 2. The L-shaped structure consists of nine bays on its vertical section, and one bay on its horizontal section, extending 90 inches and 20 inches, respectively. The shorter section is

clamped to a steel plate, which is rigidly attached to the wall. The square cross-section is 10 inches by 10 inches. Two cold air jet thrusters, located at the beam tip, serve as actuators for excitation and control. Each thruster has a maximum thrust of 2.2 lbs. Two servo accelerometers located at a corner of the square cross-section provide the in-plane tip acceleration measurements. In addition, an offset weight of 30 pounds is added to enhance the dynamic coupling between the two principal axes, and to lower the structure fundamental frequency. For identification, the truss is excited using random inputs to both thrusters. The input-output signals are sampled at 250 Hz and recorded for system identification. A data record of 2,000 points is used for identification.



First, OKID was used on the data set to obtain the initial identification with the parameter $p = 4$ in this run, i.e., the system order $p \times m = 4 \times 2 = 8$ where m is the number of outputs. Next, we applied the MIMO ARX algorithm and the MIMO block-diagonal form algorithm to the data set,

starting with the OKID result. The initial norm from the OKID result was 715.9185. After 30 iterations using the ARX algorithm, the final norm of the error was 78.3937. On the other hand, the block-diagonal form algorithm reduced the initial norm to the final value 78.86 after 30 iterations.

For the ARX algorithm, the first iteration used a step size of 0.2 and decreased the norm of the error to 585.3143. The second iteration used a step size of 0.7 and reached an error norm of 445.5199. By iteration 10 the error was 88.829. The step size was unity for all but 3 iterations with the nonunity values of 0.2, 0.7, and 0.9. The run time for this optimized identification took 8.45 minutes on a 200 megaHertz pentium PC using Matlab version 5.0.

For the block-diagonal form algorithm, the first iteration used a step size of 0.7 and decreased the norm of the error to 411.4671. The second iteration used a step size of 1.0 and reached an error norm of 203.4215. By iteration 10 the error was 81.012. The step size was unity for all but the first iteration. The run time for this optimized identification took 7.23 minutes.

Table 2 gives the OKID-identified and optimized values of the frequencies and the dampings. We may conclude that the first two modes are system modes, whereas the last two modes having very high damping ratios may be considered to be computational modes or noise modes due to measurement noise and/or system uncertainties. The inconsistency between the OKID result and the other results in damping of the fourth mode make it ambiguous whether the fourth mode is a system mode.

Table 2. Comparison of OKID-identified and optimized frequencies and dampings.

Mode No.	OKID result		ARX		Block-Diagonal Form	
	Freq (Hz)	Damping %	Freq (Hz)	Damping %	Freq (Hz)	Damping %
1	5.906	2.555	5.845	0.388	5.845	0.384
2	7.318	1.887	7.275	0.457	7.275	0.463
3	9.069	24.899	17.534	92.328	25.349	98.008
4	48.885	1.751	29.431	32.850	44.693	79.945

Let us reset the parameter $p = 40$. The OKID without any singular value truncation would then identify a model of size 80. Among the complex modes from the OKID-identified model, let us choose six complex modes as shown in Table 3 to form the reduced model for optimized identification. Before applying any optimization algorithm, it is better to improve the reduced model by using the reduced state matrix A and output matrix C to update the input matrix B and the direct transmission matrix D , and compute the initial state vector $x(0)$ (see Ref. [17]). The initial frequencies and dampings would remain unchanged in this updating process. We applied the MIMO block-diagonal algorithm to the data set, starting with the reduced OKID-identified model. The initial norm from the OKID result was 165.5505. After 17 iterations, the final norm of the error had stabilized to 6 digits and the value was 73.8241. The first iteration used a step size of 0.7 and decreased the norm of the error to 153.8757. By iteration 10 the error reduced to 81.181 which was about half of the initial error. The step size was unity for all but 5 iterations with the nonunity values ranging from 0.0002 at the last iteration to 0.9 at the 14th iteration. Other methods also produced similar results (not shown).

Examination of the identified frequencies and dampings shown in Table 3 indicates that in spite of the presence of noise and uncertainties, the OKID results are quite good when the parameter p is chosen quite large. The optimized identification improves the identification of the third mode and distinguishes the third mode from the fourth mode significantly in damping.

Table 3. Reduced OKID and optimized frequencies and dampings.

Mode No.	OKID result		Block-Diagonal Form	
	Freq (Hz)	Damping %	Freq (Hz)	Damping %
1	5.846	0.375	5.846	0.379
2	7.279	0.439	7.277	0.448
3	48.597	0.481	48.525	0.000
4	46.493	1.531	49.316	99.596
5	120.023	3.129	33.527	69.338

CONCLUSIONS

The most common objective of system identification is to produce a model from data such that when the model and the real system are subjected to the same inputs, the model outputs are as close as possible to the true system outputs. However, most system identification algorithms do not make a model that fits the data in this sense of matching outputs. In this paper we have developed algorithms that do minimize this objective function, and illustrate that the identification results fix some persistent difficulties in overestimating damping in lightly damped structures. Minimizing this output error requires minimizing a nonlinear function of many parameters. Using routine methods that are derivative free, or do not require the second derivative, produces prohibitive run times. Here we derive the needed derivative information for several classes of model realizations, and develop a nonlinear least-squares algorithm that handles indefinite second partials. Examples demonstrate that the algorithms fix biases in identification results by other methods, and that the run times for the algorithms are reasonable and practical.

REFERENCES

- [1] L. Ljung, *System Identification: Theory for the User*, Prentice Hall, Englewood Cliffs, NJ, 1987.
- [2] J.-N. Juang, M. Phan, L. G. Horta, and R. W. Longman, "Identification of Observer/Kalman Filter Markov Parameters: Theory and Experiments," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 2, March-April 1993, pp. 320-329.
- [3] J.-N. Juang, *Applied System Identification*, PTR Prentice Hall, Englewood Cliffs, NJ, 1994.
- [4] C.-W. Chen, J.-K. Huang, M. Phan and J.-N. Juang, "Integrated System Identification and Modal State Estimation for Control of Large Flexible Space Structures," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 1, January-February 1992, pp. 88-95.

- [5] M. Phan, J.-N. Juang, and R. W. Longman, "Identification of Linear Multivariable Systems by Identification of Observers with Assigned Real Eigenvalues," *Journal of the Astronautical Sciences*, Vol. 40, No. 2, April-June 1992, pp. 261-279.
- [6] M. Phan, L. G. Horta, J.-N. Juang, and R. W. Longman, "Linear System Identification Via an Asymptotically Stable Observer," *Journal of Optimization Theory and Application*, Vol. 79, No. 1, October 1993, pp. 59-86.
- [7] M. Phan, L. G. Horta, J.-N. Juang, and R. W. Longman, "Identification of Observer/Kalman Filter Markov Parameters Using Limited Experimental Data," *Proceedings of the Eighth VPI&SU Symposium on Dynamics and Control of Large Structures*, May 1991.
- [8] M. Phan, L. G. Horta, J.-N. Juang, and R. W. Longman, "Improvement of Observer/Kalman Filter Identification (OKID) by Residual Whitening," *Journal of Vibrations and Acoustics*, Vol. 117, April 1995, pp. 232-239.
- [9] R. K. Lim, M. Q. Phan, and R. W. Longman, "State Estimation with ARMarkov Models," Department of Mechanical and Aerospace Engineering, Technical Report No. 3046, Princeton University, Oct. 1998.
- [10] R. K. Lim, M. Q. Phan, and R. W. Longman, "State-Space System Identification with Identified Hankel Matrix," Department of Mechanical and Aerospace Engineering, Technical Report No. 3045, Princeton University, Sept. 1998.
- [11] J.-N. Juang, R. W. Longman, and Y.-C. Chen, "Improved Speech Coding Based on Open-Loop Parameter Estimation," to be published as a NASA Technical Memorandum.
- [12] Juang, J.-N, Cooper, J. E., and Wright J. R, "An Eigensystem Realization Algorithm Using Data Correlations (ERA/DC) for Modal Parameter Identification," *Control-Theory and Advanced Technology*, Vol. 4, No. 1, pp. 5-14, 1988.

- [13] Juang, J.-N., and Pappa, R. S., "An Eigensystem Realization Algorithm for Modal Parameter Identification and Model Reduction," *Journal of Guidance, Control and Dynamics*, Vol. 8, No. 5, Sept.-Oct. 1985, pp. 620-627.
- [14] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, London, 1981, General Reference.
- [15] Nahi, N. E., *Estimation Theory and Applications*, John Wiley and Sons, Inc., New York, 1969, Chapter 4, pp. 104-114.
- [16] R. W. Longman, M. Bergmann and J.-N. Juang, "Variance and Bias Confidence Criteria for ERA Modal Parameter Identification," *Proceedings of the AIAA/AAS Astrodynamics Conference*, Minneapolis, Minnesota, August 1988, pp. 729-739.
- [17] J.-N. Juang, "System Realization Using Information Matrix," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 3, May-June 1997, pp. 492-500.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE October 1999		3. REPORT TYPE AND DATES COVERED Technical Memorandum
4. TITLE AND SUBTITLE Optimized System Identification			5. FUNDING NUMBERS WU 632-10-14-04	
6. AUTHOR(S) Jer-Nan Juang and Richard W. Longman				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) NASA Langley Research Center Hampton, VA 23681-2199			8. PERFORMING ORGANIZATION REPORT NUMBER L-17897	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001			10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA/TM-1999-209711	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified-Unlimited Subject Category 39 Distribution: Standard Availability: NASA CASI (301) 621-0390			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) <p>In system identification, one usually cares most about finding a model whose outputs are as close as possible to the true system outputs when the same input is applied to both. However, most system identification algorithms do not minimize this output error. Often they minimize model equation error instead, as in typical least-squares fits using a finite-difference model, and it is seen here that this distinction is significant. Here, we develop a set of system identification algorithms that minimize output error for multi-input/multi-output and multi-input/single-output systems. This is done with sequential quadratic programming iterations on the nonlinear least-squares problems, with an eigendecomposition to handle indefinite second partials. This optimization minimizes a nonlinear function of many variables, and hence can converge to local minima. To handle this problem, we start the iterations from the OKID (Observer/Kalman Identification) algorithm result. Not only has OKID proved very effective in practice, it minimizes an output error of an observer which has the property that as the data set gets large, it converges to minimizing the criterion of interest here. Hence, it is a particularly good starting point for the nonlinear iterations here. Examples show that the methods developed here eliminate the bias that is often observed using any system identification methods of either over-estimating or under-estimating the damping of vibration modes in lightly damped structures.</p>				
14. SUBJECT TERMS System Identification, System Realization, Modal Testing, System Model Optimization			15. NUMBER OF PAGES 36	
			16. PRICE CODE A03	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	